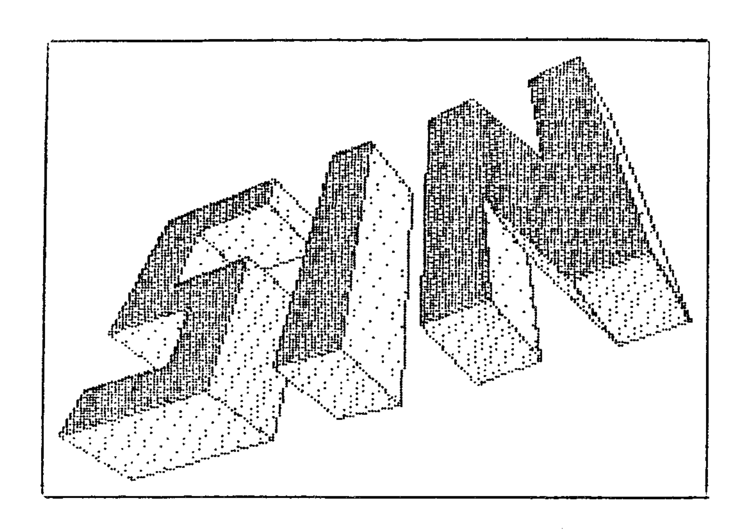


-SINCLAIR INFORMATIONS NOUVELLES-

No.6 janvier 1986 $2.00

Rédacteur: Réal Gagnon (544 Hermine #2 Québec P.Q. G1N 2G6)
Collaborateur: Patrice Giroux, Jaquelin Cauchon
Ce bulletin est conçu avec l'aide d'un traitement de texte
TASWORD II de Tasman Software (version 2068)
et d'une imprimante SMITH-CORONA, Fastext 80.

** DANS CE NUMERO **

Entre deux octets...

Avant tout j'aimerais profiter de l'occasion pour souhaiter à tout les lecteurs de SIN une année remplis de joie et avec le moins de BUGS que possible. Certaines personnes m'ont demandé comment le titre de la page converture (celui en 3d) était crée. Pour le faire j'utilise VU 3-D, de Psion. Ce logiciel permet la création d'objet en 3 dimensions et on peut déplacer ces objets dans l'espace. Ainsi pour SIN, je fais les 3 lettres et je les déplace dans l'espace, lorsque le LOOK me plait, j'utilise une option de VU 3-D pour simuler une lumière qui plongerais sur SIN.

J'aimerais rappeler également à tous que les pages de SIN vous sont toutes grandes ouvertes. Pour que SIN reste en vie, il a un besoin pressant de matériel. Le choix du sujet est entièrement à votre discrétion. Que soit pour une iniation au BASIC ou pour un programme que vous avez écrit, n'hésitez pas, écrivez-le. Et si vous avez besoin d'un traitement de texte pour le faire et bien, SIN vous fournira un !

Récemment je me suis rendu compte que plusieurs ne savent pas quels genres de programme qui sont publiés dans les gros magazines britanniques (YOUR COMPUTER,SINCLAIR USER...). Dans le prochain numéro de SIN, il y aura 2 exemples de ceux-çi, un pour le SPECTRUM et un autre pour le QL. Celui pour le SPECTRUM est un TOOLKIT pour obtenir et controler des SPRITES à partir du BASIC. C'est le meilleur et le plus facile d'utilisation que j'ai vu à date. Pour le QL, ce sera un PRINT SPOOLER, un programme faisant appel aux possibilités de MULTI-TASKING du QL. Ce programme permet au QL de faire sortir un texte sur imprimante tout en ayant complètement le controle du clavier pour continuer à programmer par exemple. Içi le QL fait 2 travail en meme temps, imprime le texte sur imprimante et continue à accepter les commandes qu'on lui envoie. Ces 2 programmes absolumment fantastiques ont été tiré de la revue YOUR COMPUTER.

SINCLAIR NEWS

DEUX de MOINS !

Le magazine américain CREATIVE COMPUTING vient de rendre l'ame. Le fondateur DAVID H. AHL était connu des "sinclairistes" car il était également le fondateur et le rédacteur en chef de la

regrettée revue SYNC. La disparition de CREATIVE COMPUTING n'est pas une véritable surprise car le contenu de la revue était en chute libre depuis plus d'un an. Le nombre de pages avait diminué de moité. Par contre, la disparition subite de POPULAR COMPUTING était plus inattendue. Ceci révèle dans quel état de crise se trouve l'industrie de la micro-informatique nord-américaine. Avec ces 2 disparitions, BYTE et COMPUTER SHOPPER sont les seuls magazines majeurs américains qui provoque en moi encore un certain intéret.

LES CLUBS AU CANADA

Il existe, en Colombie-Britannique, un groupe d'utilisateurs du SINCLAIR QL. On peut entrer en contact avec eux en écrivant à OKANAGAN QL USER GROUP, c/o RR2, SITE 79, BOX 10, SUMMERLAND , B.C. CANADA V0H 1Z0. Toujours dans la meme province, il y a un groupe pour les ordinateurs SINCLAIR en général, leur adresse est: VANCOUVER SINCLAIR USER GROUP, P.O. Box 788, New Westminster, B.C. CANADA V3L 4Z8. En Alberta, il y a le CALGARY TIMEX SINCLAIR USER GROUP (CTSUG), Box 201 STN G ,CALGARY ALBERTA CANADA T3A 2G2.

Plus près de nous, à TORONTO, il y a le TORONTO TIMEX-SINCLAIR USER GROUP, P.O. Box 7274 STN A, TORONTO CANADA M5W 1X9.

SINCLAIR FRAPPE ENCORE

Durant la nouvelle année qui débute, SINCLAIR annonce la sortie de 3 nouveaux ordinateurs. Le premier sera sans aucun doute le SPECTRUM 128K. Il en existe déjà une version en ESPAGNE. Il possède 2 modes soit le mode SPECTRUM (compatibilité oblige) et le mode 128K. Ensuite vient le PANDORA, il s'agit d'un SPECTRUM portable avec 2 microdrives intégrés. Enfin, le dernier et non le moindre ,l'ENIGMA. Construit autour d'un MC68000, l'ENIGMA aurait en standard 1 MEGABYTE de mémoire ainsi que les 4 logiciels de PSION fournis avec le QL en ROM. Egalement les microdrives seraient mis de coté, 2 lecteurs de disquette 3.5'' prendraient la relève.

Mentionnons également que SINCLAIR travaille sur un prototype de téléphone cellulaire qui serait de la grosseur d'un paquet de cigarettes, son prix : £99 .

MAGAZINES AMERICAINS

Avec la nouvelle année qui débute il serait bon de regarder les magazines américains dédiés aux ordinateurs SINCLAIR qui existent toujours. Le plus important est TS HORIZONS, l'abonnement pour le CANADA est de $21.00 US (1 an,12 nos). SUM magazine est quant à lui moins dispendieux ($18.00 US, 1 an,12 nos). TIME DESIGNS parait à tout les 2 mois, son abonnement est de $15.00 US (1 an, 6 nos). Enfin mentionnons QuarTerS qui propose une édition 4 fois par année, l'abonnement est de $8.00 US. Celui-çi est publié par WMJ DATA SYSTEMS.

TS HORIZONS , 2002 Summit Street, Portsmouth, Ohio 45662 USA
SUM , 3224 NW 30 Avenue, Gainesville, Florida 32605 USA
TIME DESIGNS, 29722 Hult Rd., Colton, Oregon 97017 USA
QUARTERS , 4 Butterfly Drive, Hauppauge, New York 11788 USA

LE TS2068, UN CP/M ?

ZEBRA SYSTEMS annonce qu'ils ont terminé l'implantation du CP/M sur le 2068. Avec le nouveau système d'exploitation le 2068 fonctionne en 64 colonnes. CP/M devrait etre disponible au début de 1986, il ne resterait qu'à signer la licence de DIGITAL RESEARCH.

ZEBRA SYSTEMS INC., 78-06 Jaimaica Avenue, Woodhaven, NY 11421

HARDWARE NEWS

LARKEN ELECTRONICS propose pour le TS2068 un contrôleur de disquette qui me semble être très intéressant. Il est heureux de noter qu'il fonctionne même lorsque le 2068 est en mode SPECTRUM (ROMSWITCH). Le système d'exploitation est sur EPROM, et il peut etre placé n'importe où en mémoire pour une plus grande compatibilité avec les logiciels existants. Comme les "sacrés" cassettes,il supporte 3 types de fichier soit BASIC, Bloc d'octets et les tableaux (DIM). Il peut charger en mémoire 32K en 7.5 secondes. Le contrôl eur accepte des unités de disquette du genre SHUGGART 455 (ou IBM compatible) double face. Ceci signifie qu'on peut utiliser des disquettes 5'' au lieu de 3.5'' ou 3'' plus difficile à trouver. Les cables sont fournis mais l'alimentation ne l'est pas,cependant LARKEN fournit les plans nécessaires pour celle-ci. L'alimentation doit fournir 5V,1A et 12V,.8A. Une garantie de 60 jours est offerte. Le prix est de $119.95 canadien plus $4.00 pour les frais de livraison.

LARKEN ELECTRONICS, RR#2 NAVAN, ONTARIO, CANADA K4B 1H9

Le décodage du clavier du 2068

Une des fonctions du BASIC du 2068 qui peut sembler mystérieuse est la fonction IN. Elle donne la valeur d'un port d'entrée/sortie spécifique. Le clavier étant un port d'entrée, nous pouvons donc le lire avec IN. Mais pourquoi utiliser IN alors que INKEY$ remplit déjà trés bien cette tâche! L'avantage de la fonction IN est que nous pouvons détecter si plusieurs touches sont appuyées en même temps.

Le clavier du 2068 est divisé en une matrice de 8 lignes et 5 colonnes. Lorsqu'une touche est appuyée, un contact se fait entre la ligne et la colonne.

Tapez ce petit programme:

```
10 PRINT AT 0,0;IN 32766;" "
20 GOTO 10
```

si aucune touche n'est appuyée les bits seront toutes à "1", ce qui équivaut à 31 en décimal. Si on appuie sur ESPACE, nous aurons 30, car la bit0 est tombée à "0" pour indiquée qu'une touche est appuyée. Si maintenant on appuie sur ESPACE et B en même temps, nous aurons 14, car la bit4 (-16) et la bit0 (-1) sont à "0".

ADRESSE DU PORT DE SORTIE		Bit à "0" si touche appuyée				
		Bit4	Bit3	Bit2	Bit1	Bit0
Dec	Hex	(-16)	(-8)	(-4)	(-2)	(-1)
32766	7FFE	B	N	M	SYMBL	ESPACE
49150	BFFE	H	J	K	L	ENTER
57342	DFFE	Y	U	I	O	P
61438	EFFE	6	7	8	9	0
63486	F7FE	5	4	3	2	1
64510	FBFE	T	R	E	W	Q
65022	FDFE	G	F	D	S	A
65278	FEFE	V	C	X	Z	CAPS

Note: La touche BREAK est en fait la même que la barre ESPACE.
Par conséquent une lecture du clavier en ASSEMBLEUR pourrait ressembler à ceci: ex. si la touche "X" est appuyée, on quitte le programme

```
        LD BC,65278     ;port d'entrée de la touche "X"
        IN A,(C)        ;lecture du port,résultat dans A
        RRA             ;met Bit2 dans le flag C
        RRA
        RRA
        JP NC,quitte    ;saut à quitte si Bit2=0 (flagC=0)
        ...suite...     ;sinon le programme se poursuit
quitte  RET             ;routine "quitte"
```

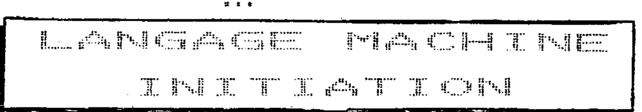
Bien sur, la ROM du 2068 contient déjà des routines qui font ce travail. Premièrement il faut noter que à chaque interruption du Z80, le 2068 scrute le clavier pour détecter si une touche est enfoncée. Si il y a en une, le code de cette touche se retrouvera dans la variable-système LASTK (23560). Mais auparavant nous devons nous assurer que la bit 5 de 23611 (FLAGS) est à "1" pour à valider notre lecture de LASTK.

Une lecture du clavier en utilisant ces variables-système pourrait ressembler à ceci:

```
        XOR A           ;met A à zéro
        BIT 5,(IY+1)    ;test la bit 5
        JR Z,aucune     ;saut si bit à "0"
        LD A,(23560)    ;A=touche qui a été
        RES 5,(IY+1)    ;  appuyée
        RET             ;on remet à "0"
aucune  ...             ;la routine continue
                         iÇi si aucune n'a
                         été appuyée.
```

Comme nous avons vue précédemment, si nous voulons détecter si BREAK (i.e. CAPS+BREAK) est appuyée, nous faisons un IN 32766 et un IN 65278 et nous testons leur bit 0 respective. Ou bien nous pouvons utiliser la routine de la ROM qui se trouve à 8201d. Au retour de cette routine le Carry Flag est à "0" si BREAK est appuyée et à "1" si non. Ainsi pour sortir d'une routine machine avec BREAK, nous faisons:

```
CALL 8201d
RET NC      ;retour au BASIC si C="0"
...
```



LANGAGE MACHINE INITIATION

Le langage machine est le seul langage que "comprend" le microprocesseur de chaque ordinateur. Pour que l'utilisation d'un langage évolué tel que le BASIC soit possible, il faut que chaque instruction soit interprété en un équivalent d'opération en langage machine. La mémoire ROM d'un ordinateur contient une série de routines qui servent entre autres à interpréter les instructions BASIC.

Chaque microprocesseur possède son propre jeu d'instructions. En ce qui nous concerne, nous allons nous attarder sur le microprocesseur Z80.

Le Z80 comprend 2 régistres simples (8 bits). Ce sont les régistres A et F. Le régistre A est le plus versatile car plusieurs instructions lui sont réservées. C'est l'Accumulateur. Le régistre F est réservé pour les indicateurs d'états.

Fig.1-Régistre F

b7				b3			b0
S	Z	-	H	-	P/V	N	C

C : Carry, dépassement ou retenue lors d'opération arithmétique.

N : Ajustement de valeur décimale. Utilisée par le CPU lors de conversion BCD en valeur décimale.

P/V: Parité -> paire=1 impair=0

- : inutilisé

H : Carry du demi-octet (4 bits)

Z : Indicateur de zéro -> Z=1 si zéro

S : Signe -> négatif=1 positif=0

En plus de ces 2 régistres, le Z80 possède 3 régistres paires (16 bits): HL, DE et BC. Ils sont représentés par 2 lettres parce qu'ils peuvent etre utilisés tous individuellement. En d'autres mots, on peut utiliser HL en un seul bloc de 16 bits ou H et L en 2 blocs de 8 bits.

Fig.2-Régistres du Z80

A	F	A'	F'
B	C	B'	C'
H	L	H'	L'
D	E	D'	E'
IX			
IY			
SP			
PC			
I	R		

Deux régistres, paires seulement, viennent s'ajouter au compte. Le IX et le IY servent au mode d'adressage indexé. Les 3 derniers sont indispensables au microprocesseur: IR, PC et SP.

IR: Instruction register, c'est le régistre d'instruction. Il contient l'instruction qui doit etre exécuté.

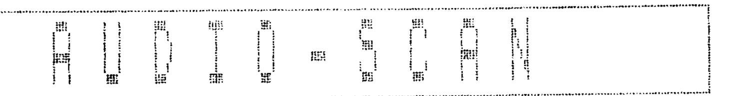
PC: Program Counter, compteur ordinal. Il contient l'adresse de la prochaine instruction. Quand une instruction est exécutée le PC augmente de 1 automatiquement.

SP: Stack Pointer, le pointeur de pile contient l'adresse du du dessus de la pile.

La pile peut se comparer à une pile d'assiettes ou les assiettes sont des octets (bytes), et le pointeur de pile indique l'assiette qui est sur le dessus; donc la première qui peut etre prise. Si on place une assiette sur la pile, le SP augmente de 1; si on enlève une assiette de sur la pile, le SP diminue de 1.

Dans le prochain article, nous verrons comment s'exécute une instruction de langage machine.

Patrice Giroux



Le programme qui suit donne une représentation visuelle de ce qui se passe à l'entrée EAR du 2068. Tapez le programme qui suit et sauvez-le sur cassette. Pour essayer le programme faites RUN et installez une cassette dans votre magnéto, pressez PLAY. Choisissez le type de représentation que vous désirez parmi les 3 disponibles.

Programme: AUDIO-SCAN

```
 100 REM ______________________ audioscan
 101 GO SUB 490
 102 DEF FN a(l)=l+ INT (.5+l/30)
 103 GO SUB 410
 105 GO TO 200
 110 DRAW  INK  FN a(l/1.5);X-N, INT (-L+Y)/1.5: LET x=n: LET y=
l: RETURN
 120 DRAW  INK  FN a(l/1.5);0,-l/2: RETURN
 200 PAPER 0: INK 7: BRIGHT 1: CLS
 210 PRINT  AT 0,10; FLASH 1;"Audio Scan"
 220 PRINT  AT 2,0;"Ce programme donne une represen-tation graph
ique entrant par la prise audio du 2068.."
 225 PRINT  AT 7,2;"Chargez une cassette ou autre   source de si
gnal par l'entre    audio et selectionnez une option"
 230 PRINT  AT 12,5;"1--graphique lineaire"; AT 13,5;"2--graphiq
ue barre"; AT 14,5;"3--plot par points"
```

```
240 INPUT  INVERSE 1;"Entrez l'option (1 a 3) ";q: IF q<1 OR q>
3 THEN  GO TO 240
250 CLS : PRINT #1; INVERSE 1;"Touche espace pour arreter scan"
: LET i$="": LET x=0: LET y=0
255 FOR n=0 TO 255: LET l= USR tone: PLOT n, INT (l/1.5)
260 IF q<3 THEN  GO SUB 100+(q*10)
265 LET i$= INKEY$ : IF i$=" " THEN  GO TO 300
270 NEXT n
275 GO TO 250
300 PRINT #1; INVERSE 1;"m=menu r=recommence f=fin        ": PAU
SE 0
310 LET i$= INKEY$ : IF i$="m" THEN  RUN 102
320 IF i$="r" THEN  GO TO 270
330 IF i$="f" THEN  STOP
340 GO TO 300
400 DATA 1,0,255,17,0,0,219,254,203,119,32,1,19,16,247,66,75,20
1
410 LET tone=65368
420 FOR n=tone TO tone+17: READ d: POKE n,d: NEXT n: RETURN
490 BORDER 5: PAPER 4: INK 1: CLS
500 PRINT  AT 2,6;"PROGRAMME AUDIOSCAN"
510 PRINT  AT 4,2;"traduit par J.Cauchon 1985"
520 PRINT  AT 8,10;"INSTRUCTIONS"
530 PAUSE 200: CLS
540 PRINT "Le programme qui suit vous offreune representation v
isuelle du  signal audio qui entre par la   prise ecouteur du TS
 2068.       Ce programme peut servir pour   evaluer ou comparer
des casset- tes avec des problemes de char- gement (LOAD), pour
un simple   amusement ou pour tout ce que   vous pourrez imagine
r."
550 PRINT : PRINT "Occasionnellement des plots pourraient etre
hors ecran, pour re-medier a ce probleme recommencezle programme
 et reduisez le ni- veau du signal sonore.            L'echelle pe
ut aussi etre changedans la ligne 102 ( DEF FN )."
560 PRINT : PRINT : PRINT "Pressez une touche pour debuter"
570 IF  INKEY$ ="" THEN  GO TO 570
580 CLS : RETURN
```

PROGRAMME: Solitaire

Voici un classique des jeux de table, le solitaire. Le but du jeu est d'éliminer le plus possible de pions, idéalement il ne doit en rester qu'un seul. On élimine un pion en sautant par dessus, soit de droite <-> gauche ou haut <-> bas, mais pas en diagonale. Premièrement, vous mettez le curseur sur la pièce que vous désirez bouger à l'aide des touches curseur (5 à 8). Une fois ceci fait, vous appuyez "0". Alors vous indiquez la direction du saut avec une des touches curseur. Si après avoir choisi un pion vous changez d'idée, vous n'avez qu'à appuyer sur "1" pour annuler le coup.

BONNE CHANCE !

```
   5 REM       SOLITAIRE    SIN#6
  10 REM
  11 BORDER 0: PAPER 0: INK 7: CLS : GO SUB 8000
  20 DIM b$(9,9): LET xb=10: LET yb=6
  22 FOR x=6 TO 16: PRINT  AT x,9; PAPER 2;"              ": NEXT
 x
  30 FOR y=1 TO 9: FOR x=1 TO 9: IF y>3 AND y<7 OR x>3 AND x<7 T
HEN  PRINT  AT y+yb,x+xb; PAPER 6; INK 0; CHR$ 147: LET b$(y,x)=
"A"
  40 NEXT x: NEXT y
  50 LET g=0: LET b$(5,5)="E": PRINT  AT 5+yb,5+xb; PAPER 6; INK
 2; CHR$ 144
  80 PLOT 87,47: DRAW 73,0: DRAW 0,73: DRAW -73,0: DRAW 0,-73
 100 LET y=11: LET x=15
 105 PRINT  AT 4,10; INVERSE 1;"MOUVEMENT ?"
 107 IF  INKEY$  <> "" THEN  GO TO 107
 120 PRINT  AT y,x; OVER 1; BRIGHT 1; INK 8; PAPER 8; FLASH 1;"
"
 130 LET a=x-xb: LET b=y-yb
 134 PRINT  AT 19,9;g;" coup";"s" AND g>1
 136 PAUSE 0: LET a$= INKEY$
 140 IF a$="" THEN  GO TO 107
 145 IF a$="0" THEN  GO TO 500
 150 IF a$="5" OR a$="8" THEN  GO TO 200
 160 IF a$ <> "6" AND a$ <> "7" THEN  BEEP .1,.1:: GO TO 110
 170 PRINT  AT y,x; PAPER 8; OVER 1; INK 8;" ": LET y=y+(a$="6"
AND y<15)-(a$="7" AND y>7)
 190 GO TO 110
 200 PRINT  AT y,x; OVER 1; PAPER 8; INK 8;" "
 210 LET x=x+(a$="8" AND x<19)-(a$="5" AND x>11): GO TO 110
 500 IF b$(b,a) <> "A" THEN  GO TO 105
 510 PRINT  AT 4,10; INVERSE 1;"DIRECTION ?"
```

```
520 LET a$= INKEY$ : IF a$="" THEN  GO TO 520
530 IF a$="1" THEN  BEEP .1,.01: GO TO 105
540 IF a$="6" AND b <= 7 THEN  GO TO 1000
550 IF a$="5" AND a >= 3 THEN  GO TO 1100
560 IF a$="7" AND b >= 3 THEN  GO TO 1200
570 IF a$="8" AND a <= 7 THEN  GO TO 1300
580 GO TO 520
1010 IF b$(b+1,a)="A" AND b$(b+2,a)="E" THEN  LET yd=1: LET xd=0
: GO TO 2000
1020 GO TO 500
1110 IF b$(b,a-1)="A" AND b$(b,a-2)="E" THEN  LET yd=0: LET xd=-
1: GO TO 2000
1120 GO TO 500
1210 IF b$(b-1,a)="A" AND b$(b-2,a)="E" THEN  LET xd=0: LET yd=-
1: GO TO 2000
1211 GO TO 500
1310 IF b$(b,a+1)="A" AND b$(b,a+2)="E" THEN  LET xd=1: LET yd=0
: GO TO 2000
1320 GO TO 500
2001 BEEP .01,.01
2010 PRINT  AT y,x; PAPER 6; INK 2; CHR$ 144: LET b$(b,a)="E": L
ET y=y+yd: LET x=x+xd: LET b=b+yd: LET a=a+xd: PRINT  AT y,x; PA
PER 6; INK 2; CHR$ 144: BEEP .01,2: LET b$(b,a)="E": LET y=y+yd:
 LET x=x+xd: LET b=b+yd: LET a=a+xd: PRINT  AT y,x; PAPER 6; INK
 0; CHR$ 147
2011 LET g=g+1
2020 LET b$(b,a)="A": LET x=x-2*xd: LET y=y-2*yd: GO TO 105
8000 LET t$="      * S O L I T A I R E *      "
8010 PRINT #0; AT 0,0;: FOR i=1 TO  LEN t$: BEEP .01, CODE t$(i)
/2
8020 PRINT #0; INVERSE 1;t$(i);: NEXT i
8025 LET t$="      TIMEX - SINCLAIR   2068    "
8030 FOR i=1 TO  LEN t$: BEEP .01, CODE t$(i)/3: PRINT #0; PAPER
 2;t$(i);: NEXT i
9000 FOR i=1 TO 2: READ a$: FOR z=0 TO 7: READ a: POKE  USR a$+z
,a: NEXT z: NEXT i
9010 DATA "d",60,66,90,90,90,66,60,0,"a",60,66,66,66,66,66,60,0
9050 PLOT 20,0: DRAW 215,0: DRAW 0,175: DRAW -215,0: DRAW 0,-175
9055 RETURN
9999 SAVE "soli" LINE 1: VERIFY ""
```

LA PAGE du Sinclair QL

Si vous êtes l'heureux propriètaire d'un QL, vous avez sûrement remarqué que le QL possède une police de caractère très complète, en effet, il est possible d'obtenir tout les caractères accentués français avec la touche CTRL. Nul besoin de redéfinir des caractères. Le tableau qui suit, résume les combinaisons nécessaires pour les caractères français.

CTRL_SHIFT_3	é
CTRL_0	è
CTRL_1	ê
CTRL_-	à
CTRL_.	â
CTRL_;	û
CTRL_8	ô
CTRL_SHIFT_9	ç

Jusqu'içi pas de problème, mais si on utilise QUILL, le traitement de texte du QL, nos caractères accentués ne sortirons pas correctement sur l'imprimante car ils ne correspondent pas à des codes ASCII standards. Heureusement le microdrive qui contient QUILL, possède également un petit programme appelé INSTALL_BAS qui va résoudre ce petit problème. Ce programme sert d'interface entre QUILL et notre imprimante. Faites LRUN MDV1_INSTALL_BAS pour charger ce programme. Choisissez l'option correspondante à votre imprimante (référez-vous à votre manuel d'utilisation, section CONCEPT vers la fin). L'exemple que nous allons faire correspond à une imprimante de type EPSON. Mettez votre curseur sur TRANSLATE1. TRANSLATE signifie TRADUIT, cette option signifie au QL que à chaque fois qu'il rencontre un certain caractère, il en envoie un autre à l'imprimante suivant une séquence déterminée. Ainsi pour le "é" nous aurons ceci:

TRANSLATE1 131,ESC,"R,SOH,123,ESC,"R,NUL

-131 est le code du QL pour le "é"

-ESC "R sélectionne un jeu de caractères internationaux de l'imprimante.

-SOH signifie que c'est le jeu de la FRANCE qui est sélectionné.

-123 est le code de "é" du jeu de la FRANCE.

-ESC "R NUL remet l'imprimante à l'état normal.

M

Faisons une autre exemple, cette fois-çi pour le "ô":

TRANSLATE8 152,ESC,"R,SOH,111,BS,94,ESC,"R,NUL

-152 est le code du QL pour le "ô"

-ESC "R SOH sélectionne le jeu de la FRANCE.

-111 est le code ASCII de "o"

-BS est une mnémonique signifiant BACK SPACE, retour à l'arrière d'un caractère.

-94 est le code ASCII de "^", comme nous sommes revenu à l'arrière avec BS, "^" serait fait sur "o" pour donner "ô".

Voici la liste de tout les TRANSLATE pour obtenir nos caractères accentués à l'imprimante de type EPSON.
N'oubliez pas de sauver le tout sur microdrive une fois le travail terminé.

TRANSLATE1	131,ESC,"R,SOH,123,ESC,"R,NUL	(é)
TRANSLATE2	144,ESC,"R,SOH,125,ESC,"R,NUL	(è)
TRANSLATE3	145,ESC,"R,SOH,101,BS,94,ESC,"R,NUL	(ê)
TRANSLATE4	141,ESC,"R,SOH,64,ESC,"R,NUL	(à)
TRANSLATE5	136,ESC,"R,SOH,93,ESC,"R,NUL	(ç)
TRANSLATE6	142,ESC,"R,SOH,97,BS,94,ESC,"R,NUL	(â)
TRANSLATE7	152,ESC,"R,SOH,111,BC,94,ESC,"R,NUL	(ô)
TRANSLATE8	155,ESC,"R,SOH,117,BS,94,ESC,"R,NUL	(û)

ASTUCE2068...TRUC&ASTUCE2068...TRUC&ASTUCE2068...TRUC&ASTUCE2068

Quand nous voulons spécifier des attributs pour notre écran, nous faisons cette longue suite de commande:

50 PAPER 1: INK 7 : FLASH 0 : BRIGHT 0 : CLS

Mais voici une méthode plus rapide et plus économe de mémoire. Nous pokons la valeur désirée dans la variable-système appelée ATTR_P qui se trouve à 23693.

Retapons maintenant notre ligne 50:

50 POKE 23693, BIN 00001111: CLS

FLASH0 BLEU
BRGHT0 BLANC

Il suffit de se rappeler que la valeur égale à PAPER*8+INK et +128 si FLASH=1 et +64 si BRIGHT=1. Ex. Pour PAPER=3, INK=6, FLASH=1 et BRIGHT=1, nous avons (3*8) + 6 + 128 + 64 = 222

PAPER INK FLASH BRGHT
3 6 1 1

Note: Pour le bas de l'écran (zone INPUT), on POKE à l'adresse 23624, les attributs se calcule de la meme façon que çi-dessus.

Fonctions Utiles: DEF FN

Parmi les instructions du SINCLAIR BASIC, il s'en trouve une qui est très puissante mais aussi très sous-utilisée. Il s'agit de DEF FN.

DEF FN signifie DEFined FuNction, il s'agit en fait d'une fonction que le programmeur peu créer. Le 2068 permet de créer jusqu'à 26 fonctions numériques et 26 fonctions alpha-numériques. Une des applications classiques de cette commande est la conversion hexadécimal-décimal. Ecrivons un programme qui convertit un nombre hexa. (<256) en son équivalent décimal.

```
 10 INPUT  "HEX:";H$:IF  LEN  H$<2
 THEN LET H$=H$+"0"
 20 GOSUB 100
 30 PRINT H$,H: GOTO 10
100 REM hex -> dec
110 LET H=16*(CODE H$(1)-48-(7 AND
 H$ (1)<"9"))+CODE H$(2)-48-(7 AND
 H$(2)>"9")
120 RETURN
```

mais si on refait ce programme en utilisant DEF FN, nous aurons un programme beaucoup plus compact.

```
10 DEF FN H(H$)=16*(CODE H$(1)-48
-(7 AND H$(1)>"9"))+CODE H$(2)-48
-(7 AND CODE H$(2)>"9")
20 INPUT "HEX:";H$:IF LEN H$<2 TH
EN LET H$=H$+"0"
30 PRINT H$,FN H(H$):GOTO 20
```

comme on peut le constater on crée la fonction avec DEF FN et on l'utilise avec FN. Une fonction peut avoir autant d'argument qu'on le désire. Ex. Calcul de la moyenne de 5 chiffres: DEF FN m(a,b,c,d,e)=(a+b+c+d+e)/5 .

Comme vous le savez, la manière dont le Z80 indique une adresse en mémoire n'est pas de tout repos, en effet une adresse occupe 2 octets, chaque octet peut contenir au maximum 255. Le Z80 place donc l'octet le moins significatif en premier et le plus significatif en second. Si nous voulons savoir, par exemple, quelle adresse pointe une variable système, disons PROG (début

des programme BASIC), nous faisons PRINT PEEK 23635+256*PEEK 23636. Mais si sous devons répeter cette opération plusieurs fois et ce, avec des variables systèmes différentes, il serait bon de définir une fonction.

L'exemple qui suit donne les adresses pointées par quelques variables systèmes:

```
 10 DEF FN A(X)=PEEK X+256*PEEK(X+1)
 20 CLS
 30 FOR I=0 TO 3
 40 READ POINTEUR,N$,E$
 50 PRINT N$;" ";E$
 60 PRINT "Adresse pointée=>",FN a(pointeur)
 80 DATA 23627,"VARS",Adresse des variables"
 90 DATA 23635,"PROG","Adresse de debut BASIC"
100 DATA 23675,"UDG","Adresse des UDGs"
110 DATA 23730,"RAMTOP","Adresse du dernier octet
 accesible au BASIC"
```

Si au lieu de lire, nous voulons écrire de nouvelles valeurs, nous pouvons utiliser ces 2 fonctions:

```
DEF FN b(x)=x-256*INT (x/256)
DEF FN h(x)=INT (x/256)
```

FN b(x) est pour l'octet le moins significatif et FN h(x) est pour le plus significatif. Ainsi si nous voulons changer l'adresse des UDGs, nous faisons:

```
10 DEF FN b(x)=x-256*INT (x/256)
20 DEF FN h(x)=INT (x/256)
30 INPUT "Nouvelle adresse des UDGs:";adr
40 POKE 23675,FN b(adr)
50 POKE 23676,FN h(adr)
```

Voici 3 autres applications possibles de DEF FN. En premier lieu, nous verrons comment transformer des coordonnées PLOT en coordonnées PRINT AT. Le programme de démonstration met un message qui FLASH à l'écran et entoure celui-Çi avec un cadre qui ne FLASH pas. Puis nous verrons comment simuler la commande

DRAW du BASIC MICROSOFT. Enfin nous utiliserons DEF FN pour changer directement les attributs à l'écran.

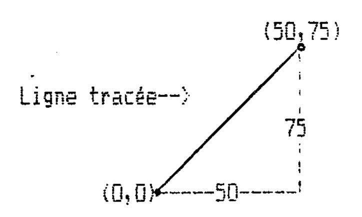
Voici 4 nouvelles fonctions qui servent à transformer des positions PLOT et positions PRINT AT et vice-versa.

```
Position PRINT AT->Position PLOT : DEF FN x(colonne)=8*colonne
                                   DEF FN y(ligne)=(21-ligne)*8
Position PLOT->Position PRINT AT : DEF FN c(posx)=INT (posx/8)
                                   DEF FN l(posy)=21-INT(posy/8)
```

Voici une application possible de 2 de ces fonctions:

```
 10 DEF FN X(C)=8*C
 20 DEF FN Y(L)=(21-L)*8
 30 LET A$="TIMEX SINCLAIR 2068"
 40 LET L=20: LET C=3
 50 PRINT AT L,C;INK 7;PAPER 5;FLASH 1;A$
 60 REM  ENCADREMENT de A$
 70 LET B=LEN A$
 80 PLOT  FN x(c)-1, FN y(L): DRAW 0,8
 90 DRAW (B*8)+1,0: DRAW 0,-9
100 DRAW -((B*8)+1),0
```

La commande DRAW du 2068 fonctionne du manière différente de la plupart des autres BASICs (eg. MICROSOFT). Nous devons spécifier le déplacement en X et en Y, ainsi si nous faisons un point à (0,0) et que nous voulons tracer une ligne jusqu'à (50,75), on doit indiquer un déplacement en X de 50 et en Y de 75.



Avec le BASIC MICROSOFT, la commande DRAW demande seulement la coordonnée du point à etre relier, dans ce cas on aurait DRAW 50,75 . Alors un ligne serait tracé entre le dernier PLOT (0,0) et la coordonnée que nous venons de spécifier.

Nous allons définir 2 fonctions pour simuler le DRAW façon MICROSOFT.

1) DEF FN x(h)=h-PEEK 23677

2) DEF FN y(l)=l-PEEK 23678

23677 et 23678 sont 2 variables-système qui nons indiquent les coordonnées du dernier point affiché à l'écran avec PLOT. Dans le programme qui suit nous allons faire une PLOT (10,5) et nous allons tracer une ligne jusqu'à (50,100) à la manière MICROSOFT.

```
10 DEF FN X(H)=H-PEEK 23677
20 DEF FN Y(L)=L-PEEK 23678
30 PLOT 10,5
40 DRAW FN X(50),FN L(100)
```

Voici un petit truc pour changer les attributs d'un endroit spécifique à l'écran. La zone mémoire qui s'occupe des attributs se trouve à 22528-23296, soit 768 octets. L'octet qui definit l'attribut trouve toute sa signification si on le regarde sous sa forme binaire:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLASH	BRGHT	Attribut:PAPER			Attribut:INK		
1/0	1/0	Vert	Rouge	Bleu	Vert	Rouge	Bleu

Le Cyan s'obtient en "mélangant" le vert et le bleu, donc en binaire -> 101 (5 décimal). Le noir est l'absence de toute couleur, donc toute les bits sont à "0" -> 000 (0 décimal), tandis que le Blanc est la présence de toute les couleurs, ->111 (7 décimal). La fonction ATTR nous retourne la valeur des attributs à une position donnée: si PRINT ATTR (10,0) nous donne 171, on peut en conclure que à la ligne 10, colonne 0, le caractère FLASH, le PAPER est CYAN et le INK est MAGENTA, car 171 en binaire est 1 0 1 0 1 0 1 1.

```
              1   0   1  0  1    0  1  1
        FLASH1    ^   PAPER CYAN  INK MAGENTA
             BRIGHT0    (5)          (3)
```

Malheureusement, le contraire n'est pas possible, on ne peut faire LET ATTR (10,0)=170 pour changer notre INK MAGENTA en un INK ROUGE. Voici une solution utilisant la puissante commande DEF FN. La fonction est DEF FN a(x,y)=22528+32*x+y ou x=ligne et y=colonne, elle retourne l'adresse correspondante dans la mémoire-attribut à une position à l'écran. Tapez le petit DEMO suivant pour en savoir plus...

```
10 DEF FN A(X,Y)=22528+32*X+Y
20 FOR I=0 TO 21
30 PRINT "********************************"
40 NEXT I
50 INPUT AT 0,0;"Ligne:";x,"Colonne:";y'
"INK:";ink,"PAPER:";paper
60 POKE  FN a(x,y),paper*8+ink
70 GOTO 50
```

Le coeur du QL: Le 68008

Il règne une grande confusion autour du QL. La question est:
Est-ce que le QL est un 32, un 16 ou un 8 bits ?
Tout dépend du point de vue. Pour l'instant disons que le microprocesseur, le MC68008, qui constitue le coeur du QL est un 8/32 bits. Mais voyons d'abord ce qu'est un microprocesseur.

Le MPU (Microprocessor Unit) est la partie la plus active d'un ordinateur. Il calcule, transfert des données en mémoire, effectue des opérations sur elles et controle la plupart des autres parties de l'ordinateur (1). Jusqu'à maintenant la plupart des ordinateurs domestiques étaient contruits autour de microprocesseur 8 bits. Mentionnons le Z80 (ZX81,TS2068,Radio Shack) et le 6502 (C64,Atari). Le 68008 du QL est ce qu'on appelle un 8/32 bits parce qu'il possède un BUS de données (2) de 8 bits et qu'il effectue ses opérations sur 32 bits. Il est similaire sur ce point au INTEL 8088 du IBM PC qui est un 8/16 bits, car il possède un BUS de données de 8 bits et effectue ses opérations sur 16 bits. Le Macintosh de APPLE est 16/32 bits avec son MC68000.

Le MC68008 est la version "économique" du MC68000. En fait la série 68000 de MOTOROLA constitue un véritable famille. Le 68000 lui-même possède un BUS de données de 16 bits, le 68008 en a un 8 bits, le 68020 quant à lui en a un de 32 bits. L'avantage du 68008 est comme nous avons dit économique car les circuits intégrés (mémoires...) qui sont autour de MPU sont fait pour des 8 bits, donc plus disponible et économique , et les connecteurs sont moins imposant (un BUS de 8 bits prend moins de place qu'un BUS de 16 bits). En fait la différence entre un 68008 et un 68000 est essentiellement HARDWARE. LOGICIELLEMENT parlant les deux sont COMPATIBLE.

Le 68008 possède 17 régistres de 32 bits (comme le 68000). Le grand avantage d'avoir de large régistre est qu'il faut moins de codes machines pour effectuer une opération. Le 68000 a un BUS d'adresse (3) de 24 bits. Donc il peut adresserjusqu'à 16 MEGA BYTES (4) ! Sur les microprocesseur 8 bits, le BUS d'adresse est de 16 bits, donc la mémoire maximum adressable directement est de 64 K. Pour en avoir plus, il faut faire appel à un système compliqué de BANK SWITCHING, et la plus plupart du temps la mémoire supplémentaire n'est pas directement accessible par le BASIC, comme c'est le cas pour les 128K de ATARI et COMMODORE. Sur le QL tout est accessible en tout temps.

Bien que le BUS de données du MC68008 soit sur 8 bits, celui-çi a été conçu comme un microprocesseur fonctionnant sur des mots de 32 bits, i.e. que son organisation interne a étudié pour permettre des opérations sur 32 bits. Le 68008 dispose de 17 régistres généraux de 32 bits et d'un régistre d'état de 16 bits. Il est à noter que ces régistres généraux servent également d'accumulateur ce qui implique que chaque régistre permet d'effectuer des opérations au niveau du bit, du digit, de l'octet, du mot 16 bits et du mot 32 bits.

Le MC68008 a 56 types d'instructions. Comte tenu des nombreux modes d'adressage, le MC68008 possède plus de 1000 INSTRUCTIONS. Comparons avec le Z80. Le Z80 possède 150 types d'instructions, et en tenant compte du mode d'adressage, le Z80 posséde 696 instructions.

Finalement, il est bon de constater que le MC68008 possède le MEME répertoire d'instructions que le MC68000. En fait, sur les 1000 instructions possibles, le 68008 en a 6 ou 7 de moins que le 68000, ces instructions ayant rapport au BUS de données, cela n'a pas de conséquence majeur du point de vue compatibilité.

(1) Le QL possède un second CPU, le INTEL i8049, qui s'occupe de la lecture du clavier, de la génération du son, de la réception RS-232C, et des fonctions de l'horloge en temps réel. Egalement 2 autres circuits spéciales à SINCLAIR sont présent. Ceux-çi s'occupent de la gestion de la mémoire, de l'écran et des microdrives, de la transmission RS-232C et de l'horloge interne.

(2) BUS de DONNEES: Partie du système qui véhicule les données à lire ou à écrire dans la mémoire ou dans les portes entrée/sortie.

(3) BUS d'ADRESSE : Partie du système qui véhicule les adresses de mémoire ou de portes entrée/sortie auxquelles il est nécessaire d' accéder.

(4) Cependant le QL n'offre que 20 lignes d'adresse, ce qui donne tout de meme 1 MEGA BYTES de mémoire directement adressable.

CARACTERES ALTERNATIFS

Encore un autre set de caractères alternatifs. Tapez le petit programme suivant et faites RUN. Soyez patient car le programmme prend un peu temps pour créer les nouveaux caractères. Un fois terminé, faites POKE 23607,249. Les nouveaux caractères devraient maintenant apparaitre à l'écran.

Pour sauver le nouveau set de caractères faites:

SAVE "car"CODE 64000,768

Pour réutiliser ces caractères faites:

CLEAR 63999: LOAD ""CODE : POKE 23607,249

Pour revenir au caractères normaux du 2068 faites:

POKE 23607,60

```
caracteres alternatifs:
                  POKE 23607,249
caracteres SINCLAIR:
                  POKE 23607,60
```

```
  10 CLEAR 63999: LET rom=15616
  15 PRINT FLASH 1;"UN INSTANT S
.V.P."
  20 FOR n=64000 TO 64767
  30 LET b=PEEK rom: POKE n,b
  40 IF b/4=INT (b/4) THEN POKE
n,b+2
  50 IF b/8=INT (b/8) THEN POKE
n,b+4
  60 IF b/16=INT (b/16) THEN POK
E n,b+8
  70 IF b/32=INT (b/32) THEN POK
E n,b+16
  80 IF b/64=INT (b/64) THEN POK
E n,b+32
  90 IF b=66 THEN POKE n,b+32
  95 IF NOT b THEN POKE n,0
 100 LET rom=rom+1: NEXT n
```

